



CS361: Assignment 6: Minimum Viable Product

Overview

Implementation begins! Starting from the requirements you defined earlier, **implement your minimum viable product** (MVP). **NOTE:** You'll be asked to show your submission to teammates.

Instructions

Complete each item below and demonstrate that you did. Please **label your responses** so graders know what's what.

1. Sprint Plan

- (a) What is your **Sprint Goal**?
- (b) Provide **screenshots**, from task management system Sprint Backlog, of your Sprint Plan: The user stories and chores you'll complete for your MVP.

Minimum requirements for MVP:

- Has UI element that will (later) trigger requesting teammate's microservice
- Reflects CSH
- Works with fake data (doesn't yet have to request/receive data from teammate's service)

Requirements for user stories:

- Each has a name
- Each uses the "As a..." format
- Each is assigned to yourself
- Each is in a Sprint Backlog category/column in your task management system
- Each has a definition of done with at least three acceptance criteria
- Each has this acceptance criterion: "Reflects CSH"
- Each has a priority
- Each has a due date

2. UML Sequence Diagramming

To help with planning implementation, create two sequence diagrams: (1) How will your individual project use a service? (2) What happens when your service is requested?

- (a) For the **first sequence diagram**, what's your **use case**?
 - (b) **Create the first sequence diagram. Requirements:**
 - **End user** must be a participant
 - Your **teammate's service** must be a participant
 - **At least 3 participants**
 - (c) For the **second sequence diagram**, what's your **use case**?
 - (d) **Create the second sequence diagram. Requirements:**
 - **Your service** must be a participant
 - Your **teammates's individual project** must be a participant
 - **At least 2 participants**
3. **Minimum Viable Product Implementation**
Implement your minimum viable product, then create a **short video** showing it (5 minutes or less). **Provide a link** to the video.

MVP requirements:

- Has **UI element** that will (later) trigger requesting teammate's service
 - **Note:** It's fine if the service is called automatically when the software opens, but there needs to be some connection between user interaction and your teammate's service.
- Reflects **CSH**
 - **Note:** In the video, clearly justify how your design reflects each CSH.
- Works with **fake data** (doesn't yet have to request/receive data from teammate's service)

Video must show that the MVP meets the requirements.

4. **GitHub**

- (a) Create a **GitHub release**. It should have a **version number**. Provide a **screen-shot** of your release. Instructions for creating release:
<https://docs.github.com/en/repositories/releasing-projects-on-github/managing-releases-in-a-repository>

Submission

PDF or Word format via Canvas. **Link to video** in the document.

If you'd like to use **Kaltura** to screen-record your video:

1. Log in at <https://media.oregonstate.edu/>
2. Add New > Recording Tools / Capture
3. If needed, download Kaltura Capture when prompted
4. Open Kaltura Capture when prompted

5. Record your video
6. "Save Upload"
7. Once video is uploaded, you should get a link to it (popup/email). Follow link.
8. On the video's page: Actions > Publish > Set to "Unlisted" > Save
9. Submit the video link with this assignment

Grading

You are responsible for satisfying all criteria listed in the Canvas rubric for this assignment. You will be able to revise this assignment if you miss points.

Questions?

Please ask via Ed so that others can benefit from the answers.